Sentence by Sentence Sentiment Analysis

Alberto Munisso

The aim of the paper is to show that sentence by sentence sentiment analysis provides sharp enough sentiment scores that can be used to prove that wheat-related sentiment correlates with wheat price.
Result robustness has been proven by Augmented Dickey Fuller tests to avoid the risk of spurious correlation.
However such finding still has much space for improvements, therefore some ideas will be discussed in the conclusive remarks.

Data

Data are composed by the articles corpus and wheat daily price index series published by the International Grains Council.

The first dataset represent wheat prices.
Wheat daily price index provided by the International Grains Council is composed by 4408 daily observations spanning from 05/01/2000 to 26/11/2016. However it has to be underlined that Saturdays and Sundays do not have any price observation.

The second dataset is composed by sentiment observations extracted from articles sentences.
In particular, 140971 articles scraped from the web on 27/11/2016 compose the articles corpus. Since such articles deal with many different topics, it has been developed a python program, called *The Reading Machine*, which processes the articles. After processing them, by topic they are classified into homogeneous clusters. After articles classification using *The Reading Machine*, 15826 agriculture-related articles are selected. In the next step 2217 articles in which appear at least once the word *"wheat"* are selected for the analysis. This dataset however is not homogeneous in time because not all days show an article containing sentences in which appear the word *"wheat"*, therefore time window 2001-2016 observations are scattered unevenly, namely they show approximately 15 day time-gaps and a big number of observations is clustered in the years 2014, 2015 and 2016.
In the next step 10875 sentences in which appear the word *"wheat"* are extracted from the 2217 articles subset: these sentences are analyzed *sentence by sentence* using an *unmodified* version of VADER analysis tool, namely compound sentiment is extracted from them. Observed *compound* sentiment is a weighted result of each sentence positivity, neutrality and negativity scores, therefore it measures the overall sentence by sentence sentiment.
However compound sentiment observations are quite *noisy*, therefore *Kalman Filter* is applied to derive *unobservable* sentiment state given *observable* sentiment noisy measures: unobservable sentiment state is called *Filtered Sentiment*.
Once *Filtered Sentiment* is computed, all *Filtered Sentiment* observations are collapsed on daily basis, namely the mean of each day *Filtered Sentiment* is computed.

As final step these two datasets are merged in a unique dataset where each entry is composed by *unevenly spaced* but *contemporaneous* daily IGC Wheat Price Index and daily *Filtered Sentiment* observations.
This final dataset is composed by 91 observations ranging from 17/01/2010 to 25/05/2016 and it's used to study Wheat Price Index and *Filtered Sentiment* correlation.

Sentence by Sentence Sentiment Extraction

If the aim is to capture wheat related sentiment, sentence by sentence sentiment extraction approach gives sharp results even using *unmodified* VADER sentiment analysis tool.
The *ratio* of this approach can be understood by considering the following trivial statement:

"The condition of wheat crops is improving, while soil used for soybeans is still in bad conditions."

This statement is composed by two opposite sentiment sentences, therefore VADER analysis tool assigns a -0.18 compound sentiment score. It should be clear that this score do not reflect reality, because it is just the result of two counterbalancing sentiments.
The statement deals with two different *entities*, *wheat* and *soybeans*, and they show opposite associated sentiment: *wheat* compound sentiment should be positive, while *soybeans* compound sentiment should be negative. In fact +0.4215 and -0.5423 are given by VADER analysis tool when analyzing separately the two sentences composing the statement.

An *entity* can be defined as something that exists by itself and it shouldn't be hard to imagine that many different *entities* are usually discussed in a single article. For example in an agricultural-related article dealing with agricultural trade liberalisation between Hungary and EU (article id: 5991) only a few statements, if not sentences, are actually dealing with wheat, while the remaining statements deal with poultry, meat *et cetera*.
In this context it's not a good strategy to extract the whole article sentiment, because the different *entities*-related sentiments will just compensate themselves in a very foggy sentiment score.
Sentence by sentence sentiment extraction solves this problem by considering just sentiment related to one pre-defined *entity* that, in this paper, is *wheat*.

In order to convince the reader about this method strenghts, a few sentences from the data set will be shown:

Sentence	***Compound Sentiment Score***	***Article ID***
The most dramatic change among flour exporters was the council forecast that shipments from Russia will fall to a minimal 10.000 tonnes in 2010-11 as a result of the government-imposed export ban in the wake of the drought-curtailed wheat crop.	*-0.5574*	*39046*
Production prospects in the rest of the wheatbelt from Queensland right around to South Australia have just received a boost from recent rains.	*0.5994*	*39067*
this agreement puts that money to work in research and development efforts for the wheat industry.	*0.4939*	*76171*
Buyers are independently tracking the wheat-corn spreads of different origins.	*0.0*	*113905*

However, even concentrating *sentence by sentence* on a single *entity*, *noisy* nature of sentiment is still present. Let's consider the following statement:

"A rainy season is going to help wheat crops, but it may be dangerous for wheat germs."

This statement is dealing with one *entity*: wheat. However wheat-related sentiment is somehow tricky to measure. The issue is given by the fact that the statement is composed by two sentences: the first one that bears positive sentiment and the second one that is connoted by negative sentiment. In other words in this sentence we observe two opposite sentiment sentences that are *observable*, but we cannot observe the real *unobservable* sentiment of the statement. The main problem is given by the fact that, in a whole corpus, a set of *observable* sentiments are usually scattered around a single *unobservable* sentiment.
In face-to-face communication the same problem is present: when someone delivers a speech, his/her real sentiment remains always at a certain degree *unobservable*, while the sentiment delivered by his/her words is *observable*. As human beings we are greatly helped in getting *unobserved* sentiment from *observed* sentiment by *empathy*, namely the aggregation of capabilities related to human sensitivity, and *memory* about that person previous statements perceived sentiment.

A Sentiment Index should measure the real *unobservable* sentiment rather than just the *observable* sentiment, but obviously it cannot rely on *empathy*. However it can rely on

memory, therefore sentiment *observable* measurements have to be filtered using the *Kalman Filter*, which gives an hint of the real *unobservable* sentiment.

The Kalman Filter

The Kalman filter is used to estimate the process (also called state) underlying a set of measurements.
Consider the following State-Space Model

$$\vec{x}_{i+1} = A\vec{x}_i + Gu_i$$

$$\vec{y}_i = H_i\vec{x}_i$$

In this model only $\vec{y}_i$ is directly observable, while $\vec{x}_i$ is unobservable: this situation leads to the Observer Design Problem, namely the basic problem of estimating an internal state of a linear system by observing just the system outputs.

Consider now that we want to estimate the following state starting from a set of measurements. The model is

1. $x_k = Ax_{k-1} + Bu_k + w_{k-1}$
2. $z_k = Hx_k + v_k$

Equation 1 describes the process, while equation 2 describes the measurements. Moreover we say that w_k, v_k are the process and measurement noise: they are assumed to be *independent*, *white* and *normally distributed*. Therefore we can define w_k, v_k probability distributions as

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

where Q, R are respectively the *process noise* and the *measurement noise* error covariances.
Moreover it has to be underlined that matrices A, H as the covariances Q, R may change at each step, but the discrete case of the Kalman filter will be considered, therefore they are assumed to be constant.

The goal of the Kalman Filter is to find the K minimizing the *a posteriori* estimate error covariance $P_k = E\left(e_k e_k^{|}\right)$, where e_k is the prediction error $e_k = x_k - \hat{x}_k$ and $\hat{x}_k$ is the *a posteriori* prediction $\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$ based on *a priori* estimates $\hat{x}_k^-$.
In the discrete case K is found by

$\min e_k = \min(x_k - \hat{x}_k)$

therefore by minimizing the prediction error is found

$$K_k = P_k^- H^{|}(HP_k^- H^{|} + R)^{-1}$$

and is used for the *a posteriori* state prediction $\hat{x}_k$, that is

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

In particular as R goes to 0, Kalman filter weights the residual $(z_k - H\hat{x}_k^-)$ more heavily, while as the *a priori* estimate error covariance P_k^- goes to 0, Kalman filter weights the residual less heavily. In other words

$$\lim_{R_k \to 0} K_k = H^{-1}$$

$$\lim_{P_k \to 0} K_k = 0$$

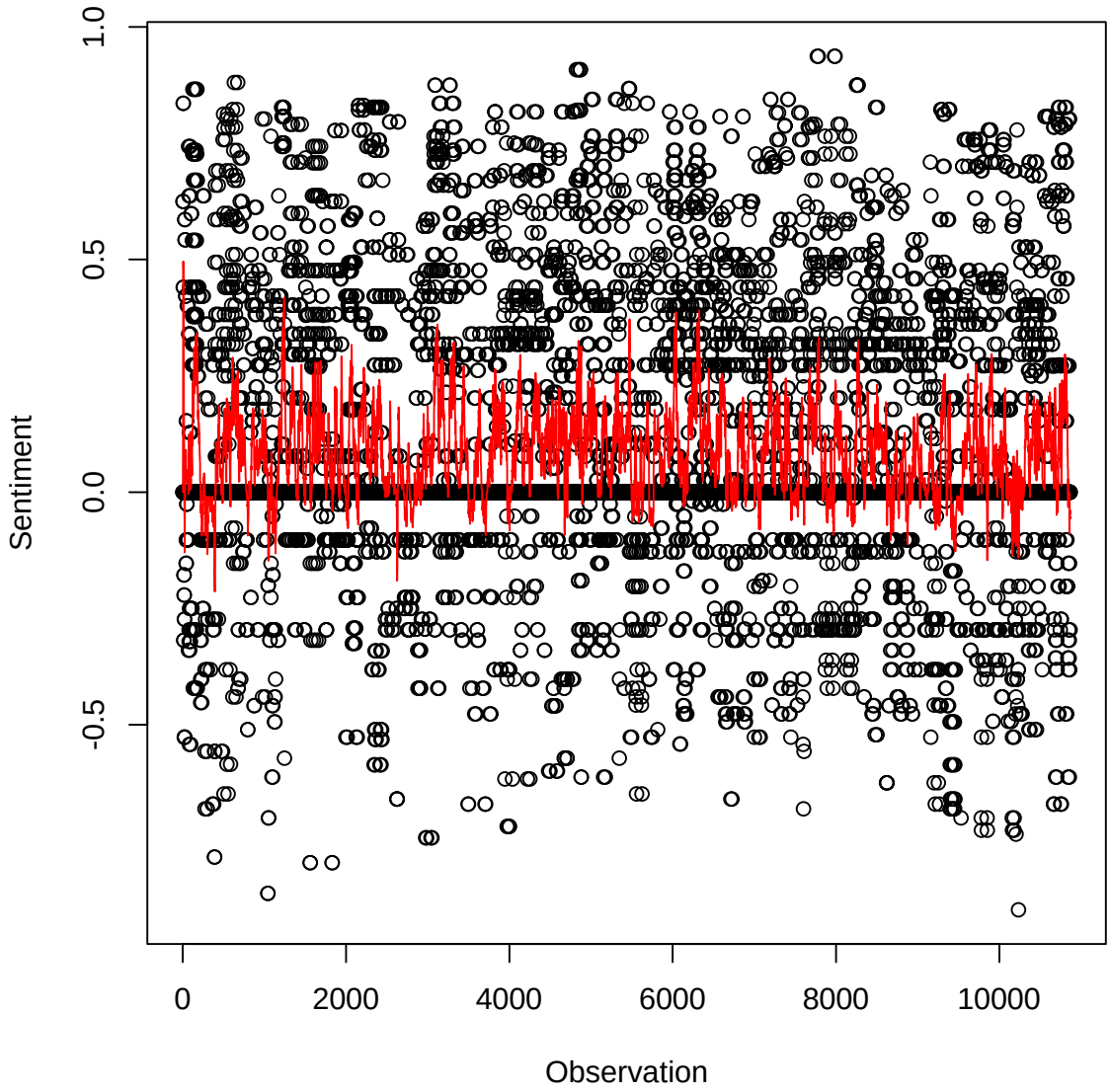
Kalman Filter algorithm is a two-step procedure where a *Time Update* is followed by a *Measurement Update*. While the *Time Update* projects forward in time the current state and error covariance estimates to obtain the *a priori* estimates for the next time step, during the *Measurement Update* a new observation is incorporated into the *a priori* estimate to obtain an improved *a posteriori* estimate. The algorithm is therefore a *predictor-error* one.

The algorithm starts after initial estimates $\hat{x}_{k-1} = x_0$ and $P_{k-1} = P_0$ are given for the first Time Update, then the following cycle is started

Time Update	***Measurement Update***
1) Project the state $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$ 2) Project the error covariance ahead $P_k^- = AP_{k-1}A^{\mid} + Q$	1) Compute Kalman K $K_k = P_k^- H^{\mid}(HP_k^- H^{\mid} + R)^{-1}$ 2) Update the estimate with observations z_K $\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$ 3) Update the error covariance $P_k = (I - K_k H)P_k^-$

In choosing the initial estimates can be said $x_0 = 0$, but $P_0 = 0$ will cause the filter believe that $x_0 = 0$. By choosing any $P_0 \neq 0$ the filter will eventually converge.

Since VADER extracted sentiment observations are *noisy*, Kalman Filter has been applied to ordered by date sentences *compound* sentiment scores.



In the plot above the dots are the single sentences compound sentiment scores, while the red line is the *Filtered Sentiment.*
It can be said that *Kalman Filter* provides an *unobservable* sentiment estimate of the unobservable process generating the *observable* sentiment scores, which in this framework are seen as noisy measurements. Under this point of view the different *observable* sentences sentiment scores are assumed to be generated around the *unobservable* sentiment state in a Gaussian way: this reflects the assumption of *white* noise errors in the measurements.

However *observed* compound sentiment scores are really sensitive to used sentiment analysis tool, namely the *dictionary*, therefore by changing or modifying the *dictionary* also the *observed* sentiment measurements will change. If *observed* sentiment measurements change by effect of used *dictionary* modifications, also *Filtered Sentiment* series change.

(Un)Modified VADER:

The unmodified version of VADER Sentiment Analysis Tool has been used for this analysis.
VADER Sentiment Analysis Tool is a software written in Python composed by a *lexicon* and a few auxiliary scripts. The tool performs a given text textual analysis and gives back four sentiment measurements: *positivity*, *neutrality*, *negativity* and *compound*.

While *positivity* and *negativity* respectively measure a text positive or negative attitude towards the same text *entity*, the *neutrality* score measures text neutrality, namely its objectivity. For example text *neutrality* will be high if colloquial words or colloquial sinctatic structures are absent.
One of VADER biggest advantages is that *neutrality* doesn't necessarily imply lack of sentiment. In a dictionary usually holds

$$S_{pos} + S_{neg} + S_{neu} = 1$$

In other words if a text shows $S_{neu} \approx 1$, also $S_{pos}, S_{neg} \approx 0$ must be true given $0 \leq S_{pos}, S_{neg}, S_{neu} \leq 1$.

To overcome this issue VADER delivers a *Compound Sentiment* score defined as

$$-1 \leq S_{com} \leq 1$$

that is a function of S_{pos}, S_{neg} and S_{neu}. In particular S_{com} sign and magnitude are defined by sign and magnitude of the difference $S_{pos} - S_{neg}$ weighted by S_{neu}, therefore if a text has an high *neutral* score and a minimal positive difference between *positivity* and *negativity*, *compound* sentiment will be positive and proportionally big to the same text *neutrality* and difference between *positivity* and *negativity*.
Having said that, it should be clear that in VADER an highly *neutral* text can still have a *positive* or *negative* attitude towards the discussed *entity*, where the text attitude is measured by *compound* sentiment.
This point is of main importance because newspaper articles are written using a highly *neutral* style, therefore using other tools *positive* or *negative* attitudes are absorbed by text *neutrality*.

How are sentiment scores computed?
Sentiment scores are computed using the *lexicon* and a few auxiliary scripts.
While the *lexicon* is a list of words with annotated *positive, neutral* and *negative* scores, the auxiliary scripts modify the resulting sentiment score by taking into account the text sinctatic structure. For example, given that the word *"good"* has an highly positive lexicon-based score, the word *"Good"* shows a *positive compound* sentiment while the negation *"No good"* doesn't bear positive *compound* sentiment.

VADER biggest flaw is that its *lexicon* is built for Twitter tweets analysis, namely it's not optimally suited for dealing with economics and agriculture related texts: *dictionaries* are *context-oriented*. In other words while very emotional sentences bear high *compound* sentiment values, neutral economic statements as *"Prices will go up."* or *"Prices will go down."* bear zero *compound* sentiment: obviously this is not true because such sentences have an *economic* sentiment full of *predictive power*. The same can be assessed for sentences dealing

with neutral supply forecasts expectations, such as "*market year 2015-16 wheat imports could possibly go as high as 3 million tonnes".*

In order to overcome this issue an extensive research of available dictionaries has been conducted. Unfortunately many *lexicons* just provide qualititave sentiment scores and/or account *neutrality* as something opposed to *positivity* and *negativity*. However, by performing a few tests, resulted that VADER *lexicon* can be modified without altering the whole algorithm framework, namely new words can be added and existing words scores can be modified.

Even if proceding *sentence by sentence* gives sharp enough results, a modified VADER version will attain even sharper results. However it has to be underlined that adding new words to VADER *lexicon* has two downfalls: on one hand the overall dictionary *equilibrium* may be distorted by the addition of more then necessary words to contextualize it to our wheat-related sentences environment, while on the other hand each new word quantitative sentiment score is obtained through some statistical effort.

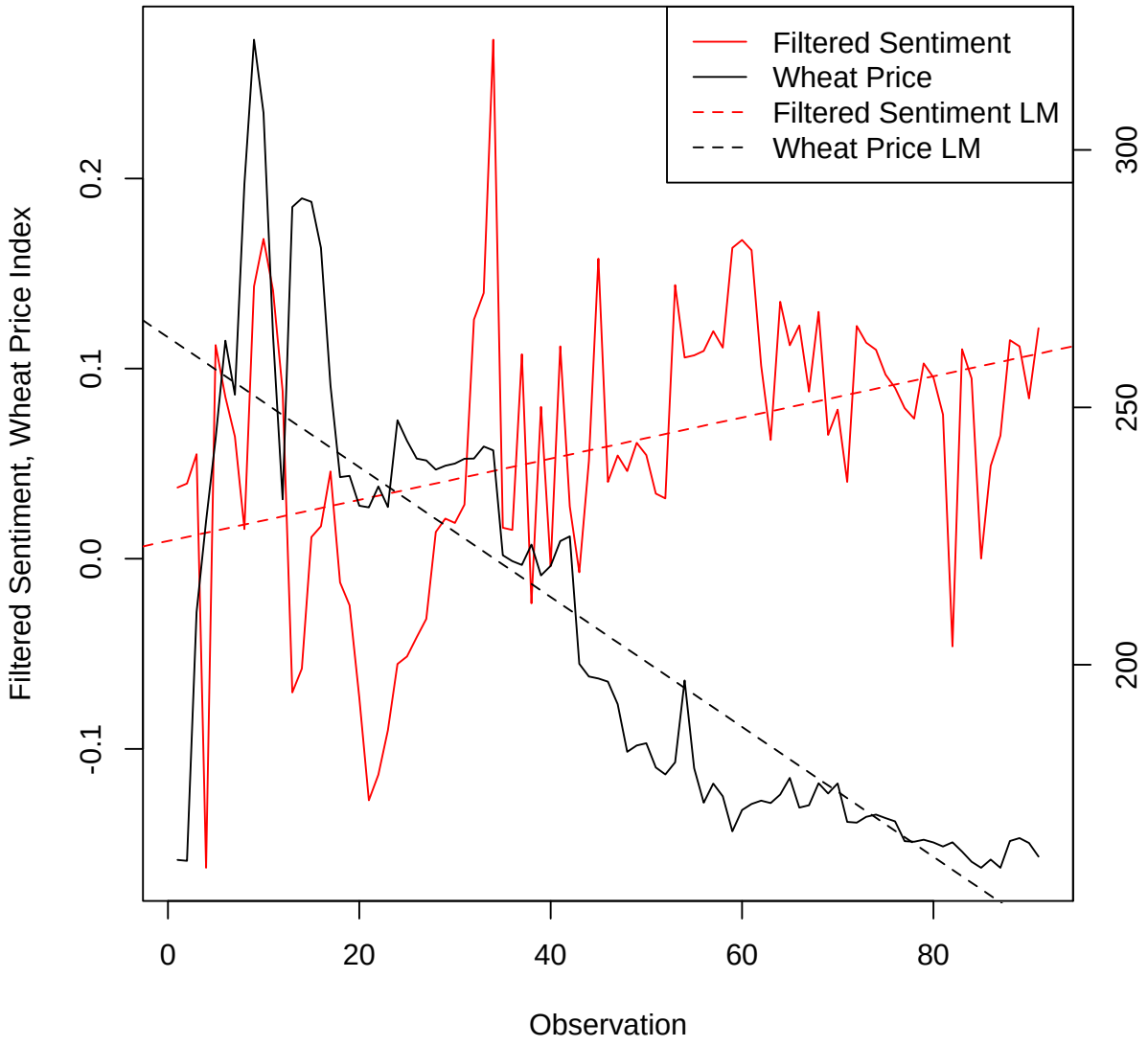
Additions to and modifications of VADER analysis tool will be discussed during next research steps, but is important to underline that *context* size is minimized by proceding *sentence by sentence* and by focusing on just one *entity*, therefore needed modifications and additions are limited. In any case, every VADER modification will be done in the scope of a *Filtered Sentiment* and Wheat Price Index correlation increase.

Having explained all the peculiarities and possible improvements of the sentiment extraction tool, analysis results can be shown.

Results

Once from the 10875 sentences corpus *compound* sentiment is extracted, 10875 Compound sentiment scores ranging from 01/06/2010 to 25/05/2016 are obtained. From them 10875 *Filtered Sentiment* estimates are computed using the *Kalman Filter*. Then the observations are aggregated into daily average observations and, after that, they are paired with contemporaneous daily wheat price observations: the result is a 91 daily Wheat Price Index and daily average *Filtered Sentiment unevenly spaced* but *contemporaneous* observations dataset. As a remark, *unevenly spaced* means that the time lag between each observation is irregular, and the most of the observations are clustered between 2014 and 2016. Prior to 2014 a very few number of observations is present, while between 2014 and 2016 there is an observation approximately every 15 days.

The plotted series are



It seems that Wheat Price Index and daily average *Filtered Sentiment* show opposite sign trends. Moreover it seems that falling prices are most of the times associated to *positive* sentiment, while rising prices are associated to *negative* sentiment.

In order to avoid the risk of *spurious correlation,* an *Augmented Dickey-Fuller Test* is performed on both the series. The tests *Lag-Order* is defined by the $AR_{(p)}$ which minimises the *Aikaike Information Criterion* for each series. Tests results are shown in the table below.

Series	***Lag-Order***	***P-Value***
$S_{Filtered}$	*2*	*0.03*
P_{Wheat}	*3*	*0.01*

For both series the *null of non-stationarity* is rejected. Series *stationarity* can be explained because, on the long run, while sentiment oscillates around zero, wheat prices are back to 2010 level. Given such tests results, it can be concluded that both the series are $I(0)$, therefore correlation analysis doesn't lead to *spurious regression*. However series *stationarity* is all a matter of perspective, therefore test results may change by changing the time window.
Once series *stationarity* have been assessed, $S_{Filtered}$, P_{Wheat} *Cross-Correlation Function* is computed.

As suspected Wheat Price Index and *Filtered Sentiment* correlation is negative: such correlation is significant at all lags and it seems that *Filtered Sentiment* lags *Wheat Price Index*. Inverse relation may be explained by reading analyzed sentences: since an unmodified VADER version has been used, many sentences have been analyzed giving positive sentiment to soil conditions improvements or trade liberalisations, while negative sentiment has been assigned to trade restrictions and threats to crop growth. Given this intuition, it appears clear that in this framework sentiment and prices are indirectly linked: sentiment is related to general conditions that have an effect on prices, namely when general conditions improve, such as a better forecasted crop growth, prices go down by the effect of an expected supply-side

positive shock. About the lagged relationship between *Filtered Sentiment* and Wheat Price Index it can be said that it may be not that significative because on one hand lags are *unevenly spaced*, while on the other hand also past lags are significant, therefore *Filtered Sentiment* seems to have some predictive power.

Conclusion

It has been shown that *Sentence by Sentence* extraction gives, thanks to an *on-single-entity* focalised approach, sharp sentiment extraction results even using an unmodified VADER version. Moreover it has been underlined that VADER sentiment analysis tool is modifiable, therefore sentiment results are completely adjustable through VADER lexicon modifications.
Since VADER extracted sentiment scores are *noisy*, a method for *unobservable* sentiment extraction from *observable* sentiment measurements has been shown: the *Kalman Filter* delivers a quite reliable *Filtered Sentiment* series that can be compared to price series.
By comparing *Filtered Sentiment* and *Wheat Price Index* it has been shown a significant negative correlation at all tested lags using an *unevenly spaced* but *contemporaneous* set of daily *Filtered Sentiment* and Wheat Price Index observations. The correlation is negative due to VADER *lexicon* peculiarities and sentences selection; such situation is given by the fact that, in the set of selected sentences, VADER gives positive scores to emphasized positive supply-sided news.

Next Research Topics:

- *Unevenly Spaced Time Series*
  Since the dataset is composed by heterogeneous time lags *contemporaneous* daily observations clustered in 2014 – 2016 time window, data are *unevenly spaced*. The issue of irregular time series lags makes many statistical tests quite unreliable. However those tests, such the *Granger Causality Test*, are useful in assessing *Filtered Sentiment* predictive power, therefore a work-around has to be found. Possible work-arounds are data interpolation, brand new set of data to add to the articles or exploitation of existing tests.

- *Articles weights*
  At the moment sentences are extracted from equally weighted articles, namely all the articles and their sources are equally important. However this assumption may be too strong: some sources are more important than others as some articles may contain more predictive power than others. An articles weighting sistem has to be developed.

- *Dictionary modifications*
  VADER Sentiment Analysis tool is modifiable through its *lexicon*. It's important to underline that this fact gives a critical upper-hand in Sentiment Index development because it permits to change resulting sentiment scores. Obviously a certain degree of dictionary consistency has to be mantained, but dictionary contextualization will greatly help in finding price and sentiment consistent relations.